Online Learning with Noise: A Kernel-Based Policy-Gradient Approach

Emmanuel Daucé, Alain Dutech

ONLINE LEARNING WITH NOISE: A KERNEL-BASED POLICY-GRADIENT APPROACH

Emmanuel Daucé
UMR 6233
Ecole Centrale de Marseille
Technopôle de Chateau Gombert
Marseille, France
edauce@centrale-marseille.fr

Alain Dutech
LORIA/INRIA - Maia Team
Campus Scientifique, BP 239
Nancy, France
alain.dutech@loria.fr

ABSTRACT

Various forms of noise are present in the brain. The role of noise in a exploration/exploitation trade-off is cast into the framework of reinforcement learning for a complex motor learning task. We present a simple and generic neuro-controler described by a linear projection from the input to the output spaces to which a gaussian independent noise is added. This stochastic controler can perform *online*-learning using a "direct policy-gradient" scheme. The reward signal is related to the sensory information, and no direct or inverse model of the system to control is needed.

The task chosen (reaching with a multi-joint arm) is redundant and non-linear. The controler inputs are projected to a feature space of higher dimension using a topographic coding based on gaussian kernels. We show it possible, through a consistent noise level, to explore the environnment so as to find good control solution. Besides, the controler is able to adapt continuously to changes in the system dynamics.

The general framework we present here should allow to build more realistic models of biological learning, in reason of its compatibility with Hebbian principles and straightforward extension to more complex stochastic neuronal models.

KEY WORDS
Reinforcement Learning, Kernel methods, Inverse Kinematics

1 Introduction

Noise, in various forms, is observed in many aspects in the brain, from spike train activity [1, 2] to large-scale variablilty in the final motor command [3]. Noise can serve many purposes like enhancement of weak signal using stochastic resonnance, facilitating the the adaptivity to changes, and inducing new exploratory behaviors. On the other hand, noise can also be seen as a perturbation that the brain must reduce in order to display stable and reliable behavior, for example though the use of population coding.

In this paper, we tackle this very general question in the light of the classical machine learning "exploration/exploitation" tradeoff. Consider an agent having to learn appropriate (i.e. most "rewarded") response in an unknown environment. Effective learning relies on the ability to use both the current knowledge to build an appropriate response, and to deviate slightly and randomly from this response in order to try new (and possibly better) combinations of commands to solve the problem. The balance between the two tendencies is delicate to find : too strong exploitation prevents to find new solutions, but too strong exploration makes the controller less reliable and increases the tendency to forget previous experience. In this setting, tuning the noise appropriately to achieve a task without ending up with an unstable system is a delicate problem.

The case of motor learning is particularly interesting to neuroscience modelling. The cardinal function of the brain is indeed to control a complex ensemble of muscles and joints on the basis of a quite unreliable set of sensory signals and effectors. Our intuition is that this intrinsic noisiness should *actively* participate in the building of new motor commands.

We tackle the problem at the level of a generic neurocontroler using a linear transformation of its inputs plus a Gaussian noise. As we will show below, this problem can be cast into the framework of reinforcement learning [4] as the noisy output neurones of the controler can be modelized whithin the family of "stochastic controlers" beeing learned in a direct "policy-gradient" scheme [5].

The global framework of reinforcement learning is a solution of choice for many problems, one of its most appealing feature being the fact that only a scalar reward signal is needed to guide learning. This reward signal is interesting from the neurological point of view : it could indeed be related to the release of specific neurotransmitters, like dopamine release observed in the cortico-striatal loops implied in the preparation and selection of appropriate motor responses [6, 7].

The model we build and simulate in this paper is not directly related on physiological data. As such, it is not aimed at modelling a specific brain function but rather at giving hints on minimal neural network settings having the capability to implement reinforcement learning using noise. We concretely implement the learning of a direct sensori-motor transformation, i.e. transforming a sensory

signal in a motor command in a closed-loop setup. The feedback is both the sensor signal itself (the consequence of the previous motor command) and a reward that relies on quantities which are directly measurable in the sensory signal itself (like visual error for instance). As such, neither motor error nor direct or inverse models of the environment are needed.

More specifically our algorithm aims at learning neuro-controlers for system with the following properties:

- **online and life-long learning**. For our controler to stay adaptive to mechanical or perceptual changes, we want an online learning scheme that can be applied in a life-long setting.
- **Continuous state and command spaces**. Although the framework of reinforcement learning is theoretically well suited to such kind of systems, pratical algorithms for this setting are still needed. Continuous command are particularly difficult to deal with.
- **Non-linear redundant system**. The system is non-linear and has more degree of freedom than the task to solve, many different solutions do exists, resulting in an ill-formed problem to solve. The task of reaching with a 4-joint arm that we consider in section 4 is a good example of such kind of tasks.
- **Model free**. Systems dynamics (*i.e.* direct or inverse models of the system) are unknown.

In reinforcement learning, dealing with continuous states and command spaces is still an open problem. A solution is to use "Actor-critic" architectures [4] extended with regression methods in order to estimate a value function over the states [8]. Recent works on "natural actor-critic" brought new impressive results. Nevertheless, Peters and Schaal [9] algorithm relies on episodic learning and Bhatnagar *et al.* online learning schemes are restricted to discreet commands.

Here, we favor the "direct policy gradient" approach proposed by Williams [5] with its REINFORCE algorithm and later developped and improved by Bartlett and Baxter [10]. The neuro-controler is expressed as a parametrized linear combination of feature functions to which an exploration noise is added. In this way, we feel that working with a well know class of function might be more tractable that trying to estimate online an *unknown* value function, a task that has many shortcoming: subsampling, overfitting, no cross-validation, no resampling...).

As detailed in section 2, the major improvement of our method over Baxter and Bartlett's binary stochastic neuro-controler is to use noisy neurons with scalar outputs. Furthermore, our neuro-controler relies on a topographic recoding of the state space, allowing it to deal with non-linear systems. As shown in section 4, our controler learns to solve closed-loop control problems, adapting to changes in the environment. As discussed in section 5, our approach is also applicable to other kind of neuro-controler, like multi-layered networks and spiking-neurons.

2 Principles

The neuro-controler we consider computes a command vector $\vec{\mathbf{u}}$ as a linear combination of features function of the input $\vec{\mathbf{x}}$. That is to say,

$$\vec{\mathbf{u}} = \boldsymbol{W}.\vec{\Phi}(\vec{\mathbf{x}}) + \vec{\eta} \tag{1}$$

where $\boldsymbol{W}$ is a matrix of parameters, $\vec{\Phi}(.)$ are features functions that allow to recode the inputs and $\vec{\eta}$ is "exploration" noise, the importance of which is explained below. *In this section*, for simplicity, the feature function will be the identity and thus $\vec{\Phi}(\vec{\mathbf{x}}) = \vec{\mathbf{x}}$.

The objective is then to find the "best" neurocontroller, i.e. the set of parameters $\boldsymbol{W}$ which optimizes an objective function J of the rewards r_t received by the neurocontrolers at instants (t) . The algorithm is based on a gradient descent in the space of parameters along the gradient of J according to $\boldsymbol{W}$.

More formally, let us consider that, for a state $\vec{\mathbf{x}}$, a command $\vec{\mathbf{u}}$ is chosen according to a density of probability $q(\vec{\mathbf{x}}, \vec{\mathbf{u}}, \boldsymbol{W})$. If the reward received at time t is r_t, the objective function for an horizon of T is:

$$J = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r_t\right]. \tag{2}$$

Then, it has been shown [10, 11] that the gradient of J can be rewritten in term of the expectation of the gradient of the logarithm of the policy, as

$$\nabla_{\boldsymbol{W}} J = \mathbb{E}\left[\frac{1}{T}\sum_{k=1}^{T} r(\vec{\mathbf{x}}_k)\sum_{i=0}^{k-1}\nabla_{\boldsymbol{W}}\log q(\vec{\mathbf{x}}_i, \vec{\mathbf{u}}_i, \boldsymbol{W})\right]. \tag{3}$$

When the exploration noise $\vec{\eta}$ is a gaussian multivariate noise, the probability density $q(.)$ of chosing a command $\mathbf{u}$ is

$$q(\vec{\mathbf{x}}, \vec{\mathbf{u}}, \boldsymbol{W}) \sim \vec{\mathcal{N}}(\vec{\mathbf{u}} - \boldsymbol{W}.\vec{\mathbf{x}}, \Sigma) \tag{4}$$

$$= \frac{e^{\left(-\frac{1}{2}(\vec{\mathbf{u}}-\boldsymbol{W}.\vec{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\vec{\mathbf{u}}-\boldsymbol{W}.\vec{\mathbf{x}})\right)}}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}}. \tag{5}$$

where D is the dimension of the command vector. Then, we have

$$\nabla_{\boldsymbol{W}} \log q(\vec{\mathbf{x}}, \vec{\mathbf{u}}, \boldsymbol{W}) \tag{6}$$

$$= \nabla_{\boldsymbol{W}}\left[-\frac{1}{2}(\vec{\mathbf{u}} - \boldsymbol{W}.\vec{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\vec{\mathbf{u}} - \boldsymbol{W}.\vec{\mathbf{x}})\right] \tag{7}$$

$$= \boldsymbol{\Sigma}^{-1}(\vec{\mathbf{u}} - \boldsymbol{W}.\vec{\mathbf{x}})\vec{\mathbf{x}}^\top. \tag{8}$$

Estimating the gradient of the objective function relates to estimating the expectation of the scalar product of the estimation noise and the inputs.

In accordance with the online stochastic approximation of the gradient used in the OLPOMDP algorithm of

Baxter and Bartlett [11], our algorithm maintains a "trace" of the gradient so as to estimate its expectation consistently with eq.(3). Using a gradient descent approach, we modify step by step the value of parameters $\boldsymbol{W}$ by a fraction of the estimate of the gradient (where α is the "learning rate"). We end up with the following algorithm. Repeat, for $k \in 1, ..., T$:

1. For the state $\vec{\mathbf{x}}_k$, compute $\vec{\mathbf{u}}_k$ according to $q(\vec{\mathbf{x}}_k, \vec{\mathbf{u}}_k, \boldsymbol{W}_k)$;
2. Read reward r_k (possibly null) and new state $\vec{\mathbf{x}}_{k+1}$;
3. Update the traces and the weights:
   (a) $z_k = \beta z_k + (1-\beta)\boldsymbol{\Sigma}^{-1}(\vec{\mathbf{u}}_k - \boldsymbol{W}.\vec{\mathbf{x}}_k)\vec{\mathbf{x}}_k^\top$
   (b) $\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \alpha r_k z_k$

with $\beta \in [0, 1[$ where $\frac{1}{1-\beta}$ defines the "width" of the trace.

In that framework, the noise is one of the main force behind learning as $(\vec{\mathbf{u}}_k - \boldsymbol{W}.\vec{\mathbf{x}}_k)$ is exactly the exploration noise $\vec{\eta}$ added to the linear combination of the inputs. Through learning, the update of the parameters are proportionnal to $\vec{\eta}.\vec{\mathbf{x}}_k^\top$.

3 Topographic recoding of the input

The learning algorithm we derive shows many similarities with a linear perceptron at the difference that the "real" error is unknown and replaced by the product of the reward and the noise. As such, our approach shares the same limitations as the perceptron regarding the class of problem it can solves. Even though a formulation of the policy-gradient for some class of multi-layer perceptron has been hinted by Williams [5], it is limited in its applicability and an exploration noise can only be added to output neurons. A potentially better alternative, quite classical, is to recode the inputs into a feature space of higher dimension.

Topographically organized dynamical systems as models of short term memory have been introduced in neuronal modeling with the Neural Field of Amari [12]. This model approximates the neuron indexes (corresponding to a position in the map) as a continuous dimension. For their simplicity and robustness, such maps have a wide range of applications in robotics and control [13].

Apart from engineering applications, an important interest has emerged about topologically organized models of short term memory in the cortex since the experiment of Funahashi et al. [14]. Some studies have established the links between biology and neural map type models, interpreted as a mean-field approximation of neuronal activity[15], and pointed out some interesting properties relating to the adaptive sharpening where the input orientation is weakly contrasted.

In our settings, we do not explicitly model the dynamics of bubble formation, but rather "recode" the external sensory cues (like target position) in the form of a topographic map of activation.

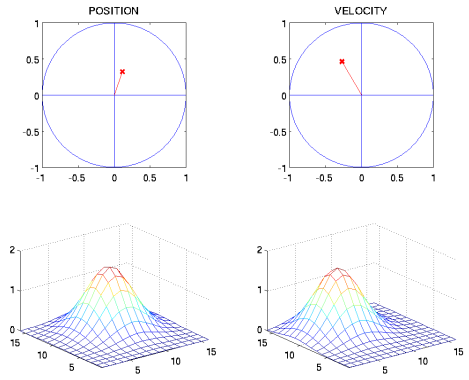

Figure 1. Gaussian-Kernel-based topographic coding.

We use a "topographic coding", redundant and smooth for better generalization, by projecting inputs into gaussian fields in the following way. Input neurons are organized in m-dimension fields of size N with coordinates $\vec{\mathrm{p}}_0, \cdots, \vec{\mathrm{p}}_N)$ so that the state of the system, or a sub-part of it, can be cast on this fields where the activity of an input neurone i is $\frac{\exp(||\vec{\mathbf{x}}-\vec{\mathrm{p}}_i||^2)}{\sqrt{2\pi}\sigma_i}$.

Figure 1 shows an example of a recoding of a 2D state (position and velocity) to two different 16×16 fields of neurones (512 dimensions).

After recoding the inputs, if we call $\vec{\Phi}(.)$ the vector of input recoding, the update rule of step (3) of the algorithm presented above becomes

3. Update the traces and the weights:
   (a) $z_k = \beta z_k + (1-\beta)\boldsymbol{\Sigma}^{-1}(\vec{\mathbf{u}}_k - \boldsymbol{W}.\vec{\Phi}(\vec{\mathbf{x}}_k))\vec{\Phi}(\vec{\mathbf{x}}_k)^\top$
   (b) $\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \alpha r_k z_k$

4 Experiments

The task is to control an arm composed of D segments on the basis of visual and proprioceptive signals, respectively in 2 and D dimension spaces. For $d \in \{1, ..., D\}$, each segment is of length ℓ_d. Consider (x_0, y_0) the coordinates of the first joint, then for d in $1, ..., D$: $(x_d, y_d) = (x_{d-1} + \ell_d \cos(\theta_d), y_{d-1} + \ell_d \sin(\theta_d))$ where $\theta_d \in [-\pi, \pi]$ is the angular direction of the d^{th} joint. The end-position of the arm $\vec{\mathbf{x}}_D = (x_D, y_D)$ is thus given by the combination of joint angles $(\theta_1, ..., \theta_D)$. Note that for $D \geq 2$, several combinations of angles give the same end-point. The coding of a coordinate in term of segments and joint angles is strongly redundant.

The end-point of the arm must reach a target specified by its location $\vec{\mathbf{x}}_C = (\hat{x}, \hat{y})$ in the visual field. This task is prototypal of an inverse kinematic problem in control theory. As said earlier, our approach does not need to learn an

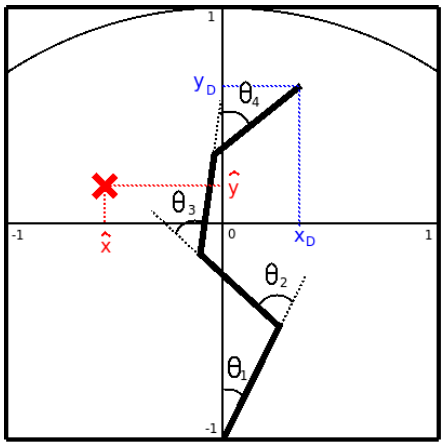

Figure 2. Multi-joint arm control setup

inverse model as the error in sensor space (visual distance between the target and the current end-point of the arm) can be used as a basis for the reward signal.

The reward we use is in fact

$$r(t) = -0.1 \left(||\vec{\mathbf{x}}_C - \vec{\mathbf{x}}_D||_2^{\,2} \right. \quad (9)$$

$$\left. +0.025 \times ||\dot{\vec{\theta}}||_2^{\,2} + 0.01 \times ||\vec{\theta}||_2^{\,2} \right) \quad (10)$$

where $\vec{\mathbf{x}}_C$ is the target position, $\vec{\mathbf{x}}_D$ the current end-position of the arm, $\vec{\theta}$ the vector of joint angle of the arm. The second line of the reward is inspired by Todorov and Jordan [16] work on modelling human gestures.

The inputs of the neuro-controler are

- the target $\vec{\mathbf{x}}_C$ recoded on a 16×16 fields of gaussian kernels with a radius of 0.5 ;
- the joints angles $\vec{\theta}$, each component of the vector is recoded on a one dimensionnal field of 16 gaussian kernels.

With a 4-joint arm, the feature space has a dimension of 320.

Every 50ms, the controler must issue a command made of a vector of joint velocities. The arm is then moved and the above reward given to the controler which must learn to reach the target. Every 4s (*i.e.*, every 80 iterations of the controler), a new target is randomly chosen.

The output is composed of 4 units, which as usual receive the linear combination of the input with weights matrix $\boldsymbol{W}$, plus a Gaussian noise $\vec{\eta}$ whose standard deviation is small ($\sigma = 0.003$).

The reward is sent every time step, and the trace is updated with $\beta = 0.9$ (time constant of 500 ms for the trace). The learning parameter α is taken such that $\frac{\alpha}{\sigma^2} = 0.01$ where σ is the standard deviation of the noise injected into the system. The weights are initially 0.

Figure 3 gives the cumulative loss (the opposite of the reward) during a session lasting 12×10^6 time steps. The level of noise remains constant as well as the learning rate (the system is continuously learning). The slope of the cumulative loss decreases and stabilizes at a value corresponding to a good achievement of the task (the arm smoothly moves from one target to the other and stabilizes on it). Note that with our setting, the loss can not be zero. At $t = 9.3 \times 10^6$ time steps, we 'block' the third joint angle at $\theta_3 = 0$. This prevents the controller from reaching the targets, and an increase in penalty (decrease in mean reward) is observed, which is progressively compensated so that the controller can reach the target anew with new combinations of commands on the remaining joint angles (fig. 4). It must be noticed that the achievement of the task is quite good despite the fact that the loss never reaches zero. In particular, the arm starts its goal-oriented movement immediately after the target has jumped at a new position, with speed decreasing with the distance to target.

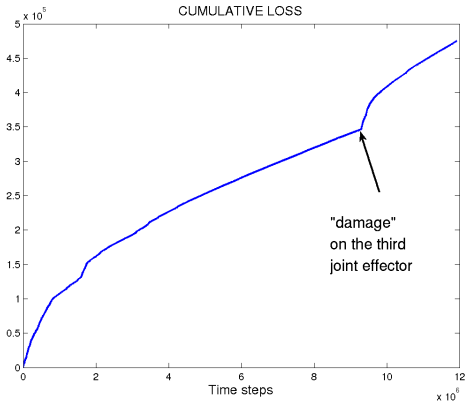

Figure 3. Cumulative loss during learning. A damage is caused on the device at $t = 9.3 \times 10^6$

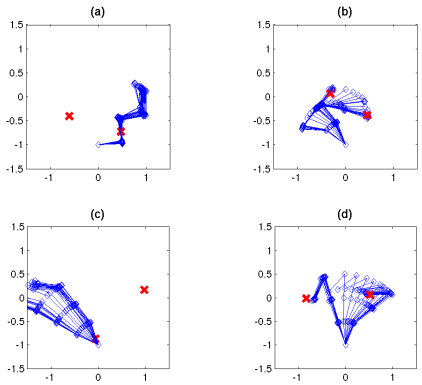

Figure 4. Typical motor responses during target switch (targets reprsented by red crosses), arm position represented on 30 time steps (a) initially (b) after learning (c) after damage (third joint is blocked at 0) (d) after recovery.

5 Discussion

First, despite the difficulty of the task and setup, our online gradient algorithm is able to derive a valid and pertinent controller. As pointed out by Baxter and Bartlett, this is not something easy to obtain [11]. With our settings, improvement of the politics is ensured at the condition of small changes (the noise level and learning coefficients are small so that the learning sessions are quite long). Faster and still reliable convergence is an objective that could be attained in episodic learning, so as to get better estimate of the gradient. In a perpective of autonomy however, online adaptation to new constraints is an important property that can not be maintain in an episodic setup without explicit failure detection (as learning is separated from exploitation in an episodic approach).

The reward used in the closed-loop setting can have many different formulations. The one we used is quite informative and gives good results, but it should be theoretically possible to learn a good controller with a more basic formulation, e.g. giving only a non-nul reward in the vicinity of the target. Another interesting point could be to explore how the formulation of the reward influence the shape, the speed or the timing of the movement. Some general properties of human gestures, like isochrony or bell-shaped speed curve [17, 18], might be measured.

Coming back to the use of noise in driving the learning, some points are to be highlighted. Noise can truly be the driving exploratroy force in learning a neuro-controler. We noted that learning was quite slow and, consequently, our future work will investigate the relation bewteen the level of noise and the "convergence speed" of the system, maybe leading to an argument in favor of more finely and adaptively tuning the noise during learning tasks. Besides, the noise we have used has independant component on each output neurone and it would be interesting to use different kinf of noise, like "prior" in the noise, "correlated" noise, "spatial" noise.

Furthermore, using a gaussian noise as the underlying probability of the outputs neurons can be interpreted as looking for a good controller in L^2-norm. As shown by [19], using the L^1-norm is more adapted when the objective is to discriminate features in the input signal. Selecting relevant commands in the case of stongly redundant command space is an interesting perspective that could be tested using an exponential distribution in the stochastic nodes of the controller.

The policy-gradient learning rule uses only 'local' quantities and as such can be formulated as a derivative of the classical Hebbian rule. Provided the neuron model is noisy, and the firing probability well-known (e.g. described as a function of the membrane potential), the resulting learning rule is perfectly local and "balanced" (the potentiation effect compensates the depression effect). This property was already pointed out by Williams in his seminal paper, in the case of simple binary stochastic neurons. Nuerous extensions to more realistic spiking neurons have been proposed since : Seung [20] treated the case of stochastic synapses with poisson firing. Baras [21] considers a Spike Response Model (SRM) [22] with a stochastic spike emission mechanism (proportional to the weighted sum of pre-synaptic activities). Florian [23] considers the more classical case of stochastic SRM neurons with escape noise, applied to a closed-loop reinforcement task. In all of those models the characteristics of the noise is mixed with the model of spike emission. An explicit decoupling of the noise term from the spike emission process is proposed in [24] using a specific temporally correlated noise source, allowing significant improement in the learning speed and effectiveness.

6 Conclusion

In this paper, we apply the reinforcement learning framework, generally devoted to difficult but discrete control problems, to tasks with continuous state *and* action spaces. Simple "perceptron-like" neuronal architecture are derived from the direct Policy-gradient approach. Inspired by observations on "neural field" topographic activity in the brain, our input are encoded into fields of gaussian kernels, allowing to bypass the limitation of linear regressor. The learning rule, driven by an exploration noise, is guided by a reward signal that is based on abailable sensory information only.

An instance of this architecture has been applied to learn a visual feedback controler for a non-linear redundant multi-joint arm. There is no need to learn any direct or inverse model of the kinematics of the system. We show in particular how a reward, directly derived from the sensory signal, can be used to learn in motor space.

We have shown that noise is a necessary component of learning in our framework in order to explore new solutions around the current one. Although, mathematically speaking, such noise must have some characteristics (for example, the noise at a node must be independant from the others), we aim at trying different kind of noise in our futur developments. One might think of less independant noise to guide or constraint the exploration. Another possibility could be to use exponential noise so as to promote feature selection inside the controler.

Furthermore, this framework is compatible with more realistic biological modelling, and can be implemented using various models of neurons as long as they remain stochastic. We recently proposed a specific implementation in [24] on the basis of SRM neurons with escape noise, and consider shaping appropriate noise as a promising direction for improving and speeding-up the learning processes, in a more biologically-realistic fashion.

References

[1] M. Shalden and W. Newsome. The variable discharge of cortical neurons: Implocations for connectivity, computa-

tion and information coding. *Journal of Neuroscience*, 18(10):3870–3896, 1998.

[2] W. Softky and C. Koch. The high irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *Journal of Neuroscience*, 13(1):334–450, 1993.

[3] C. Harris and D. Wolpert. Signal dependant noise determines motor planning. *Nature*, 394:780–784, 1998.

[4] R. Sutton and A. Barto. *Reinforcement Learning*. Bradford Book, MIT Press, Cambridge, MA, 1998.

[5] Ronald Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[6] P. Redgrave. Basal ganglia. *Scholarpedia*, 2(26):1825, 2007.

[7] Y Takahashi, G Schoenbaum, and Y Niv. Silencing the critics : understanding the effects of cocaine sensitization on dorsolateral and ventral striatum in the context of an actor/critic model. *Frontiers in Neuroscience*, 2:86–99, 2008.

[8] K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245, 2000.

[9] J. Peters and S. Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, March 2008.

[10] J. Baxter and P. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

[11] J. Baxter, P. Bartlett, and L. Weaver. Experiments with infinite-horizon, policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:351–381, 2001.

[12] S. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.

[13] G. Schöner, M. Dose, and C. Engels. Dynamics of behavior: theory and applications for autonomous robot architectures. *Robotics and Autonomous System*, 16(2-4):213–245, December 1995.

[14] S. Funahashi, C. J. Bruce, and P. S. Goldman-Rakic. Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *J. Neurophysiol.*, 61:331–349, 1989.

[15] R. Ben-Yishai, R. Lev Bar-Or, and H. Sompolinsky. Theory of orientation tuning in visual cortex. *Proc. Nat. Acad. Sci. USA*, 92:3844–3848, 1995.

[16] E. Todorov and M. I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235, 2002.

[17] P. Viviani and R. Schneider. A developmental study of the relationship between geometry and kinematics in drawing movements. *Journal of Experimental Psychology*, 17(1):198–218, 1991.

[18] T. Flash and N. Hogan. The coordination of arm movements : An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703, 1985.

[19] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on*, 50(10):2231–2242, 2004.

[20] Xiaohui Xie and H. Sebastian Seung. Learning in neural networks by reinforcement of irregular spiking. *Physical Review E*, 69, 2004.

[21] Dorit Baras and Ron Meir. Reinforcement learning, spike time dependent plasticity and the bcm rule. *Neural Computation*, 19(8):2245–2279, 2007.

[22] W. Gerstner and W. Kistler. *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

[23] Razvan V. Florian. A reinforcement learning algorithm for spiking neural networks. In *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'05)*, pages 299–306, 2005.

[24] E. Daucé. A model of cell specialization using a hebbian policy-gradient approach with "slow" noise. In C. Alippi, M.M. Polycarpou, C. Panayiotou, and G. Ellinas, editors, *Proceedings, Part I the 19th International conference on artificial neural networks (ICANN 2009)*, pages 218–228, Limassol, Cyprus, 2009.